

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-161195

(43)Date of publication of application : 23.06.1995

(51)Int.Cl.

G11C 16/06
G06F 12/00

(21)Application number : 06-165859

(71)Applicant : INTEL CORP

(22)Date of filing : 27.06.1994

(72)Inventor : FANDRICH MICKEY L
DURANTE RICHARD J
UNDERWOOD KEITH F
ROZMAN RODNEY R

(30)Priority

Priority number : 93 86186 Priority date : 30.06.1993 Priority country : US

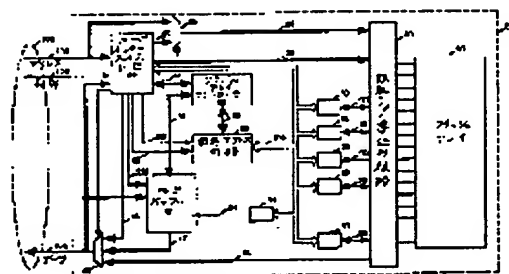
(54) FLASH MEMORY

(57)Abstract:

PURPOSE: To provide a chip mechanism extremely displaying flexibility in the case of being used in an application and controlling a flash memory increasing the entirely throughput of executing operation.

CONSTITUTION: The flash memory 310 is provided with a user interface 40 and a flash array controller 50. The user interface 40 is provided with a function receiving a user demand issued from a processor and holding together plural demands to be executed. Further, the user interface is also provided with the function controlling the priority of the demand to be executed.

The operation on the flash array such as programming and erasing or the like is executed with the array controller 50. A array controller 40 is a general processor having a program memory programmable by the user. One or plural algorithms capable of executing with the array controller are stored in the program memory. The algorithm is selected in accordance with the demand received with the user interface 40. Since each array



BEST AVAILABLE COPY

is allowed to operate in parallel, the operation input is increased.

LEGAL STATUS

[Date of request for examination] 18.05.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-161195

(43) 公開日 平成7年(1995)6月23日

(51) Int.Cl.⁵

G 1 1 C 16/06

G 0 6 F 12/00

識別記号

庁内整理番号

F I

技術表示箇所

5 5 0 A 9366-5B

G 1 1 C 17/ 00

3 0 9 Z

審査請求 未請求 請求項の数2 F D (全 20 頁)

(21) 出願番号 特願平6-165859

(22) 出願日 平成6年(1994)6月27日

(31) 優先権主張番号 0 8 6 1 8 6

(32) 優先日 1993年6月30日

(33) 優先権主張国 米国 (U S)

(71) 出願人 591003943

インテル・コーポレーション

アメリカ合衆国 95052 カリフォルニア

州・サンタクララ・ミッション カレッジ

ブーレバード・2200

(72) 発明者 ミッキー・リー・フランドリッチ

アメリカ合衆国 95667 カリフォルニア

州・プレイサーヴィル・パニング ウェ

イ・234

(74) 代理人 弁理士 山川 政樹

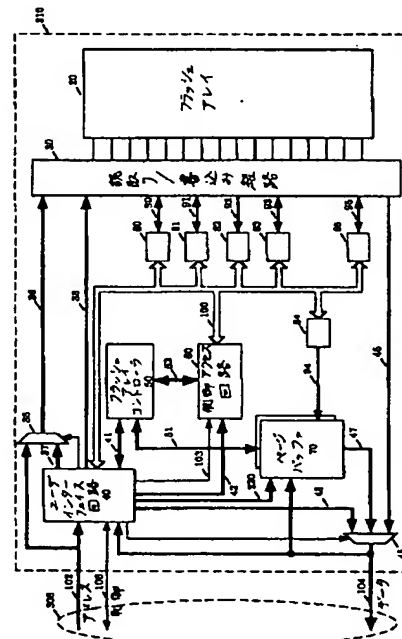
最終頁に続く

(54) 【発明の名称】 フラッシュメモリ

(57) 【要約】 (修正有)

【目的】 アプリケーションで使用する際に劇的な柔軟性を提供し、行う動作の全体的なスループットを増大するフラッシュメモリを制御するチップメカニズムを提供する。

【構成】 フラッシュメモリシステム310にはユーザインターフェイス40とアレイ制御器50を含める。ユーザインターフェイスはプロセッサにより出されたユーザコマンドを受取り、実行するため複数のコマンドを待ち合わせさせる機能を有している。ユーザインターフェイスは更に実行するコマンドの優先順位を制御する調整器としても機能する。アレイ制御器はプログラムや消去などのフラッシュアレイ上での動作を行う。アレイ制御器はユーザがプログラム可能なプログラムメモリを有する汎用プロセッサである。プログラムメモリはアレイ制御器で実行できる1つないし複数のアルゴリズムを格納する。アルゴリズムはユーザインターフェイスで受け取ったコマンドに従って選択する。



【特許請求の範囲】

【請求項1】 バスによりメモリに接続されているプロセッサを搭載したシステムのそのバスに接続されたフラッシュメモリアレイにおいて、フラッシュメモリにより実行するコマンドを受け取るユーザインターフェイスであって、実行している現在コマンドを含め1つないし複数のコマンドを格納するコマンド待ち行列を有するユーザインターフェイスと、前記ユーザインターフェイスに接続されてコマンドを受け取って実行し、アレいの読取り、消去、プログラムを行う電源を制御するアレイコントローラと、前記アレイコントローラに接続されてアレイコントローラの状態を記憶する複数のレジスタと、設定されるとコマンドの実行中に割込みの発行を可能にする割込み状態ビットとからなり、割込み状態ビットが設定され、コマンド待ち行列が実行している現在コマンドと実行する次のコマンドを含む場合、前記ユーザインターフェイスは割込みをアレイコントローラに出し、前記アレイコントローラはアレイを所定の状態にし、アレイコントローラの状態を保存し、次のコマンドを実行し、レジスタに記憶された状態を回復し、現在コマンドの実行を続行し、割り込みが、フラッシュアレイを変改することなしにフラッシュメモリシステムによって処理される、前記バスに接続された前記フラッシュメモリ・アレイシステム。

【請求項2】 バスによりメモリに接続されているプロセッサを搭載したシステムそのバスに接続されたフラッシュメモリアレイにおいて、データを記憶する複数のフラッシュセルからなるフラッシュアレイと、フラッシュメモリシステムにより実行するコマンドを受け取るユーザインターフェイスであって、実行している現在コマンドを含め1つないし複数のコマンドを格納するコマンド待ち行列を有し、かつ実行したときにコマンドを行うコードの位置を識別するコマンドアドレスを生成するユーザインターフェイスと、前記ユーザインターフェイスに接続されてコマンドアドレスを受け取るアレイコントローラであって、マイクロコントローラとプログラムメモリとを有し、前記プログラムメモリはユーザがプログラム可能で、前記コマンドアドレスは前記マイクロコントローラが実行するプログラムメモリ内のコードの位置を識別し、前記マイクロコントローラはプログラムメモリ内のコードに従ってフラッシュアレイ上で書き込み、消去動作を行うアレイコントローラと、からなり、プログラムして前記マイクロコントローラにより行う複数の機能を行うことができ、柔軟性を増大してプロセッサからのフラッシュアレイの制御をオフロードする前記バスに接続されたフラッシュメモリシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はフラッシュメモリに関し、特にフラッシュメモリ動作を制御する装置と方法に関する。

【0002】

【従来の技術】 従来の不揮発性半導体メモリの1つのタイプにフラッシュ電氣的消去、書き込み可能読取り専用メモリ（「フラッシュ」）がある。フラッシュメモリは電気信号で書き込み可能で、一旦書き込むとフラッシュメモリはそのデータが消去されるまでそれを保持する。消去後、フラッシュメモリには新しいコードないしデータを書き込むことができる。

【0003】 フラッシュメモリは従来の電氣的消去、書き込み可能読取り専用メモリ（「EEPROM」）とは消去に関して異なっている。従来のEEPROMは一般に個々のバイトの消去制御にセレクトトランジスタを使用する。他方、フラッシュメモリは一般に単一のトランジスタセルではかき高い密度を達成する。従来のフラッシュメモリの消去方法では、メモリアレイ内の全てのメモリセルのソースに高電圧を同時に供給し、それにより全アレいの消去をもたらしている。一般に論理1は、ビットセルのフローティングゲートに電子がほんのわずかし（電子があれば）格納されないことを意味し、論理0は多くの電子がブロックセルのフローティングゲートに格納されていることを意味する。フラッシュメモリの消去により論理1が各々のビットセルに格納される。そのフラッシュメモリの各々の単一のビットセルは先に消去しなければ論理0から論理1に重ね書きすることはできない。しかしそのフラッシュメモリの各々の単一ブロックセルは、それが消去状態に固有の数の電子を含むフローティングゲートに単に電子を追加するだけであるので、論理1から論理0には重ね書きすることができる。

【0004】 消去、書き込み、妥当性検査過程は、それらのステップを行うのに必要な電圧の慎重な制御を要する。例えば従来のフラッシュメモリに、256キロビットフラッシュメモリであるインテル社（カリフォルニア州サンタクララ）から市販されている28F256相補形金属酸化半導体（CMOS）フラッシュメモリがある。フラッシュメモリを制御するため、メモリには電氣的な消去と再書き込みを管理するコマンドレジスタを含める。コマンドは消去のために、制御マイクロプロセッサから標準のマイクロプロセッサ書き込みタイミングを用いて書き込まれる。コマンドレジスタの内容は、消去、書き込み回路を制御する内部状態マシンへの入力としての役割をする。

【0005】

【発明が解決しようとする課題】 フラッシュメモリの密度が増大したことで、フラッシュメモリを含むアプリケーションも増大した。それらのアプリケーションは劇的に変化し、しばしば異なる処理、制御メカニズムを必要

とする。一般にアプリケーションはバスを通してフラッシュメモリに接続されているマイクロコントローラにより駆動する。しかしマイクロコントローラでのオーバーヘッドが大きいとフラッシュ処理のスループットが減少する。更にフラッシュメモリの適切な信頼できる動作を確保するため、消去手順は厳密に守らなければならない。従ってマイクロコントローラを用いてバス上でフラッシュメモリを制御することとマイクロコントローラを通してユーザとフラッシュメモリをインターフェイスすることは、例えばマイクロコントローラにより出されている制御信号を一時的に停止する割込みその他の動作により生じる時期を逸した制御信号によるフラッシュメモリの過剰消去により生じるフラッシュエラーをもたらす可能性が増大する。

【0006】さらに、フラッシュメモリでは一般に一時に1つの動作しか行うことができない。従って低優先度の動作を実行していて高優先度の動作を実行しようとする場合、高優先度動作はその動作を行う前に低優先度動作が完了するのを待たなければならない。更にフラッシュメモリのコストが低下すると共にフラッシュメモリを利用したアプリケーションが増大している。従って様々なアプリケーションに対して柔軟性があり容易に適應できるフラッシュメモリとインターフェイスを提供することが望ましい。従って本発明の目的は、本発明の装置と方法によりアプリケーションで使用する際に劇的な柔軟性を提供し、行う動作の全体的なスループットを増大するフラッシュメモリを制御するオンチップメカニズムを提供することである。本発明の別の目的は、マイクロコントローラのオーバーヘッドを最小にし、例えばフラッシュアレイの過剰消去により生じるエラーの危険性をなくすことである。

【0007】

【課題を解決するための手段】本発明の装置は互いに共に作動して全機能性とフラッシュアレイの制御を提供するいくつかの主要要素から構成する。メモリアレイと同一構成部分にある回路は、例えばマイクロコントローラからバスを通してアドレス、コマンド、データ情報を受け取るユーザインターフェイスからなり、コマンドをアレイコントローラに出し、アレイコントローラはユーザインターフェイスから出されたコマンドで識別される消去、書き込み動作を行う。アレイコントローラはフラッシュアレイと特定の動作を行うのに必要な電圧をユーザインターフェイスとは独立した形で制御する。従ってユーザインターフェイスは様々なユーザコマンドを受け取り、アレイコントローラに実行するために送るコマンドシーケンスを制御でき、大きな柔軟性が達成できる。命令パイプライン化が可能で最大のスループットがもたらされる。更にアレイコントローラはプログラムメモリに格納されたコードに基づいて動作を行う。従ってメモリ内に格納されたマイクロコードを単に変更するだけでハ

ードウェアを変更することなしに新しいコマンドを追加したり古いコマンドを変えることができる。

【0008】回路は更にアレイコントローラにより実行されるコマンドのコンテキスト切替えを行う能力も有している。第2の過程が完了すると実行の所定の「安全」時点で実行を被割込み過程に戻し、それによりアレイ内のデータが変改されないように現在実行している過程の別の過程による安全な割込みを可能にする革新的な割込みメカニズムを提供する。例えばアレイの1つのバンクに作用する低速、低優先度の動作を一時停止し、アレイコントローラ内でコンテキスト切替えを行って高優先度の命令を実行することができる。従ってアレイの1つないし複数のバンクの消去などの時間のかかる動作を実行しているとき、アレイコントローラが消去命令と高優先度命令のコンテキスト切替えを行って消去中の書き込みのような動作を実行可能にできる次の命令をユーザインターフェイスはアレイコントローラに出すことができる。

【0009】

【実施例】以下の説明では本発明の完全な理解を提供するため数々の詳細を述べるが、当業者にはそれらの特定の詳細は本発明を実施するのに必要でないことが明かであろう。他に、よく知られた電氣的構造や回路は本発明を不必要に曖昧にしないためにブロック図形式で示すことにする。

【0010】図1はコンピュータシステム300のブロック図である。コンピュータシステム300は中央演算処理装置(CPU)302と、主記憶装置サブシステム304と、1組のフラッシュメモリ装置310-314とからなる。CPU302はユーザバス306を通して主記憶装置サブシステム304及びフラッシュメモリ装置310-314と交信する。フラッシュメモリ装置310-314はコンピュータシステム300のためのランダムアクセス不揮発性大型データ記憶装置を提供する。CPU302はユーザバス306を通して読取り記憶サイクルを生成してフラッシュメモリ装置310-314の内容を読み取る。またユーザバス306を通してフラッシュメモリ装置310-314に書き込みコマンドと書き込みデータブロックを転送してフラッシュメモリ装置310-314に書き込みを行う。

【0011】図2に本発明のフラッシュメモリシステム310のブロック図を示す。フラッシュメモリシステム310はフラッシュセルアレイ20と、ユーザインターフェイス回路40と、アレイコントローラ50と、ページバッファ回路70と、制御レジスタ回路80-85と、読取り/書き込み経路回路30とからなる。フラッシュセルアレイ20はランダムアクセス不揮発性大型データ記憶装置を提供する。1つの実施例では、フラッシュセルアレイ20は1組の32個のフラッシュアレイブロックとして構成される。フラッシュメモリシステム310はユーザバス306に接続されて示されている。ユーザバス306はユーザアドレスバス102と、ユーザデータバス104と、ユーザ制御バス106

とからなる。ユーザインターフェイス回路40はマイクロコントローラなどのマスタ装置からデータ、コマンド、アドレス情報を受け取り、その情報をバッファに格納する。バッファはそれぞれ好適には複数バッファとしてマイクロコントローラからの複数の要求を受け取り、ユーザインターフェイス回路40で考察できるようにすることに留意する。後述するようにアレイドコントローラ50の状態に基づいて即時の実行を受けるために特定のコマンドを高い優先位置に置き、アレイドコントローラ50が現在実行しているコマンドとコンテキストを切り換える。ユーザインターフェイス回路40は更にアレイドコントローラ50から状態情報を受け取り、ユーザインターフェイス回路40はそれを続いてマイクロコントローラに与える。

【0012】ユーザインターフェイス回路40はユーザバス306上で通信動作要求を受け取って処理することでユーザバス306を通してフラッシュセルアレイド20のアクセスを可能にする。ユーザインターフェイス回路40で動作要求を受け取ると、ユーザバス306から受け取ったアドレス、データ、コマンド情報はユーザインターフェイス回路40にあるバッファに格納される。要求動作がプログラムあるいは消去コマンドなどのアレイドコントローラ動作である場合、ユーザインターフェイス回路40が受け取ったコマンドやデータは続いてアレイドコントローラ50に転送され、待ち行列バス41を通して要求コマンドが実行される。アドレス情報はユーザインターフェイス回路40によりマルチプレクサ35を通して読取り／書込み経路回路30へ送られ、アドレス回線36を通して交信したアドレスでコマンドが実行される。読取り動作はアレイドコントローラが作動していないときに可能である。ユーザインターフェイス回路40はアドレスを読取り／書込み経路回路30へ送り、出力マルチプレクサをセットしてアレイドから読み取ったデータを出力する。

【0013】ユーザインターフェイス回路40とアレイドコントローラ50は両構成要素からアクセス可能なページバッファ回路70を共有し、フラッシュメモリシステムに様々な形で作動する機能を提供する。例えばページバッファ回路70は十分なデータが蓄積されてデータがフラッシュセルアレイド20に流れることができるようにフラッシュセルアレイド20に書き込むデータをバッファするのに使用することができ、フラッシュセルアレイド20でのスループットの増大をもたらす。

【0014】このバッファはユーザインターフェイス回路40とアレイドコントローラ50の両方からアクセスできる。例えばページバッファ回路70はフラッシュアレイドへの高速書込み動作を扱うのに用いる。データは引続きのアレイド書込み動作のためにユーザインターフェイス回路40によりページバッファ回路70にロードする。2つのページバッファ回路70にあるデータをアレイドへ書き込むユーザコマンドは、ユーザインターフェイス回路40の待ち行列構造を通して他のアレイドコントローラコマンドと同

様な形で待ち合わせさせることができる。そこでアレイドはページバッファ回路70に保持されたデータを用いてこれらのコマンドを実行することでプログラムすることができる。このようにユーザはページバッファ回路70をロードし、ページバッファ回路70の内容を利用するアレイドコントローラコマンドを出し、第1のアレイドコントローラコマンドが出された後にユーザインターフェイスが利用できるようにする第2のページバッファをロードすることができる。

【0015】実施例では、ページバッファはSRAMセルの128 x 19 x 2メモリアレイとする。これはいくつかの作動モードを有し、アレイドコントローラとユーザインターフェイスによりアクセスできる。モードにより8ビットないし19ビットメモリとして構成する。またモードはそれを1つの連続メモリ面ないし2つのメモリ面に制御できる。各々の面は同時にアクセス可能である。テストモードでないとき、ページバッファは2つのSRAMメモリの面に分割するのが好ましい。この2面アーキテクチャによりアレイドコントローラはメモリの半分に対する読取り／書込みアクセスを持ち、ユーザインターフェイスは残りの半分に対するアクセスを同時に持つことができる。アクセス可能な半分は固定されておらず、むしろ面の所有権を切り換えることができる。例えばこれはユーザインターフェイスがデータのページをアレイドコントローラに送り、アレイドのページプログラミングに使用することができる。ユーザインターフェイスは最初にページバッファの1つの面をデータで満たし、次にコマンド（ジャンプオフセット）をアレイドコントローラに出してページプログラムを行う。内部論理と制御信号を使用することを通して適切なコマンドを受け取ると、アレイドコントローラはホストCPUがちょうどロードした面を指定し、ユーザインターフェイスは他の面を制御してデータを書き込む。従って別々にアクセスできる2つのページバッファがあり、ページバッファプログラミングはパイプライン化してフラッシュアレイドへのデータ流れを達成してデータスループットを増加できるようにシーケンス化できる。

【0016】ページバッファはアレイドコントローラアルゴリズムにより様々な目的に使用することができる。例えばマルチブロック消去アルゴリズムは、その動作中に割込みを可能にするようにページバッファを用いてマルチブロック消去情報を記憶することができる。更にアレイドコントローラはページバッファを用いてパイプラインプログラミングのためにジャンプオフセット、データ、アドレス情報を記憶することができる。外部のユーザは更に、ユーザインターフェイスを通してページバッファをそれ自身高速読取り／書込みメモリとして使用することができる。テストモード中、アレイドコントローラにページバッファに記憶された命令を実行するように命令することができる。従ってページバッファはアレイド

ローラのマイクロコード記憶装置としての役割を果たすことができる。このモードの時、ページバッファのアーキテクチャは好適に1つの連続メモリに再構成される。

【0017】アレイコントローラ50は中央制御バス 100を通してアレイ20のプログラム、消去、検証を行うのに必要な異なる構成要素を制御する。アレイコントローラ50はフラッシュセルアレイ20に対してプログラム、消去その他の動作を行うための専用縮小命令セットプロセッサである。アレイコントローラ50には論理演算装置、汎用レジスタ、制御記憶装置、制御シーケンサが含まれている。アレイコントローラ50は待ち行列バス41を通して受け取った情報を用いてプログラムメモリの適切な場所にアクセスして動作を行うのに必要な命令を実行する。アレイコントローラ50はフラッシュセルアレイ20のフラッシュセルに対して電荷を加え、またフラッシュセルアレイ20のフラッシュセルから電荷を除去するために読取り／書込み経路回路30の高電圧回路を順序づける実施アルゴリズムを実行する。アレイコントローラ50は高電圧回路を制御し中央制御バス 100を通して制御レジスタ回路80-85にアクセスしてフラッシュセルアレイ20をアドレス指定する。

【0018】読取り／書込み経路回路30はフラッシュセルアレイ20にアクセスする読取り、書込み経路回路からなる。特に読取り／書込み経路回路30には消去機能用の適切な電圧レベルをフラッシュセルアレイ20に加えるソーススイッチ回路が含まれている。読取り／書込み経路回路30には更に、プログラム機能中にフラッシュセルアレイ20のビットラインに対してプログラムレベル電圧を駆動するプログラムロード回路が含まれている。制御レジスタ回路80-85は読取り／書込み経路回路30に対して制御信号を出す専用制御レジスタと関連回路のセットを含んでいる。専用制御レジスタは中央制御バス 100を通して書込み、読取りが行われる。

【0019】本実施例では、制御アクセス回路60によりユーザインターフェイス回路40とアレイコントローラ50の両方は中央制御バス 100を通して制御レジスタ回路80-85にアクセスすることができる。フラッシュメモリシステム 310の通常モード中、アレイコントローラ50は制御アクセス回路60を制御して中央制御バス 100を通して制御レジスタ回路80-85にアクセスする。

【0020】本実施例では、アレイコントローラ50は書込み制御信号とレジスタアドレスを対応する書込みデータと共にバス52を通して制御アクセス回路60に転送して専用制御レジスタに書込みを行う。そこで制御アクセス回路60は中央制御バス 100を通して書込みサイクルを生成してアドレス指定専用制御レジスタに書込みを行う。またアレイコントローラ50はレジスタアドレスと読取り制御信号をバス52を通して制御アクセス回路60に転送して専用制御レジスタの読取りを行う。そこで制御アクセス回路60は中央制御バス 100を通して読取りアクセスサ

イクルを生成してアドレス指定専用制御レジスタの読取りを行う。

【0021】例えば制御レジスタ回路80は1組の制御信号90に従って読取り／書込み経路回路30の高電圧回路を制御する専用制御レジスタと回路とを含んでいる。制御レジスタ回路81は1組の制御信号91に従って読取り／書込み経路回路30の専用行アクセス回路を制御する制御レジスタと回路を含んでいる。制御レジスタ回路82は読取り／書込み経路回路30からの1組の状態信号92を感知しラッチする1組の読取り専用レジスタを含んでいる。制御レジスタ回路83は1組の制御信号93に従って読取り／書込み経路回路30の読取り経路を制御する制御レジスタと回路とを含んでいる。制御レジスタ回路84はページバッファ回路70の1組のテストモードを制御するレジスタを含んでいる。制御レジスタ回路85は1組の制御信号95に従ってフラッシュメモリシステム 310の特殊テスト機能を制御するレジスタを含んでいる。

【0022】ユーザインターフェイス回路40は入力アドレスマルチプレクサ35を制御して読取り／書込み経路回路30に対する入力アドレス36を選択する。選択した入力アドレス36はユーザアドレスバス上でTTLバッファ（図示せず）により感知されたアドレスないしユーザインターフェイス回路40からのラッチされたアドレス37である。入力アドレス36は制御レジスタ回路84内の制御レジスタをプログラムすることでオーバーライドすることができる。ユーザインターフェイス回路40は出力データマルチプレクサ45を制御してユーザデータバス 104を通して出力データ転送用のソースを選択する。選択された出力データは読取り／書込み経路回路30からのフラッシュアレイデータ46、ページバッファ回路70からのページバッファデータ47ないしユーザインターフェイス回路40内に含まれる1組のブロック状態レジスタからのブロック状態レジスタ（BSR）データ48となる。このように要求装置は、アレイからのデータ並びにフラッシュメモリシステム 310の状態に関する状態情報を受け取ることができる。

【0023】従ってCPU 302はユーザ制御バス 106を通して読取りサイクルを合図する間、アドレスをユーザアドレスバス 102に転送してフラッシュセルアレイ20を読み取る。ユーザインターフェイス回路40はその読取りサイクルを検出し、入力アドレスマルチプレクサ35にユーザアドレスバス 102からのアドレスを読取り／書込み経路回路30のx、y復号回路に転送させる。またユーザインターフェイス回路40は出力データマルチプレクサ45にユーザデータバス 104を通して読取り／書込み経路回路30からのアドレス指定された読取りデータを転送させる。

【0024】CPU 302はユーザバス 306を通して書込みサイクルを生成してプログラムコマンドとデータをユーザインターフェイス回路40に転送してフラッシュセル

アレイドロにデータを書き込む。ユーザインターフェイス回路40はプログラムコマンドを検証し、プログラムコマンドとアドレス及びデータパラメータをアレイドロ50に待ち行列をつくる。アレイドロ50は指定されたデータを指定アドレスのフラッシュセルアレイドロ20にプログラムすることでプログラム動作を行う。

【0025】CPU 302はオプション的にユーザバス 306を通して書き込みサイクルを生成してプログラムデータをページバッファ回路70に転送してフラッシュセルアレイドロ20へデータを書き込む。次にCPU 302はプログラムをページバッファコマンドと共にユーザインターフェイス回路40に転送する。ユーザインターフェイス回路40はページバッファコマンドの付いたプログラムを検証してアレイドロ50に待ち行列を作る。アレイドロ50はページバッファ回路70からのプログラムデータを読み取りフラッシュセルアレイドロ20へプログラムデータをプログラムすることでページバッファコマンドの付いたプログラムを実行する。

【0026】ページバッファ回路70は2つの別々のスタティックランダムアクセスメモリ（SRAM）面からなる。2つのSRAM面は面0と面1とからなる。ユーザインターフェイス回路40はアレイドロ50により処理されたユーザコマンドに対して面0及び面1ページバッファ資源を割り当てる。更にユーザインターフェイス回路40は面0及び面1のページバッファ資源をユーザアクセスに割り当てる。ページバッファ資源面0、1は限定資源とも称する。

【0027】ユーザインターフェイス回路40は32のブロック状態レジスタ（BSR）を含んでいる。各々のBSRはフラッシュセルアレイドロ20のブロックの1つに対応している。アレイドロ50はブロック状態レジスタ内の状態ビットを維持してフラッシュセルアレイドロ20の各々のブロックの状態を示す。CPU 302はユーザバス306を通してBSRの内容を読み取る。

【0028】ユーザインターフェイス回路40はコマンドを出すユーザないしマイクロプロセッサと選択されたコマンドを実行するアレイドロ50の間の調整器として機能する。例えばアレイドロ50の現在状態を想定してユーザの要求動作が妥当かどうかを判定する。ユーザインターフェイス回路40はユーザバス 306からコマンド及びアドレス情報を入力として受取り、アレイドロ50が実行すべき動作を決定する。更にユーザインターフェイス回路40はアドレスとデータ、状態レジスタへのユーザアクセス、及び出力マルチプレクサ45の待ち行列を制御する。図3にユーザインターフェイス回路40のブロック図を示す。

【0029】図3で、コマンド状態マシン 200はCPU 302がユーザバス 306を通して提示するコマンド要求に基づいて行う動作を決定する。コマンドのタイプにはアレイドロ動作並びに状態レジスタ、テストモード、ページバ

ッファ動作がある。コマンド状態マシン 200は回路で符号化されたアルゴリズムを通して、正当なコマンドシーケンス、間違ったコマンドシーケンスからユーザを締め出す方法、アレイドロ50を開始したり中断する方法を知っている。そこで動作をアレイドロ50により行う場合はコマンドをアレイドロ50に送る。読み取り動作を行う場合、コマンド状態マシン 200はアドレスを読み取り／書き込み経路回路30に送る。コマンド状態マシン 200は出力マルチプレクサ（45、図2）を通して出力で得られるデータと入力マルチプレクサ（35、図2）を通して入力で得られるデータを制御する。更にコマンド状態マシン 200は状態レジスタ 260へのアクセスを制御する。従ってコマンド状態マシン 200はユーザバス 306を通して受けるユーザのコマンド入力を理解し、残りのフラッシュメモリシステム 310に対してユーザのコマンド入力を実行するために行うべきステップを指令する回路である。

【0030】要求されたコマンドが例えばアレイドロ読み取りないし状態レジスタ読み取りコマンドなどのアレイドロ50により実行されないものである場合、ユーザインターフェイス回路40は適切な制御信号を出して要求されたコマンドの実行を行うようにする。受け取ったコマンド要求がアレイドロ50により実行されるものである場合、ユーザインターフェイス回路40が受け取ったコマンド要求を示すコマンドコードを用いてユーザインターフェイス回路40内にあるアレイドロ50ジャンプ表（図示せず）へのインデックスを生成する。受け取るコマンドにより（例えば図4～図14を参照のこと）、ジャンプ表に格納されコマンドコードに対して写像されるオフセットベクトルはアレイドロ50に送られ、アレイドロ50はオフセットベクトルを用いてコマンドを実行するためアレイドロ50が行うアルゴリズムに対するコードをアドレス指定する。好適にはオフセットベクトルはプログラムメモリの最初の32位置の1つを引用して実際のプログラムメモリアドレスを判定して実行を開始する。本実施例では、4つの異なるクラスの 128までのコマンドを受けてユーザインターフェイス回路40で処理することができる。

【0031】本実施例では、ジャンプ表は複数コマンドを単一のオフセットベクトルに写像するように構成する。オフセットベクトルの二重性により使用するハードウェア構成により区別される複数のコマンドクラスを同一アレイドロ50アルゴリズムに写像することができる。従って複数ハードウェア構成をユーザインターフェイス回路40ないしアレイドロ50の変更なしに支援することができる。これに関してコマンドコード変換メカニズムは、アレイドロ50に交信する適切なハードウェア制御信号とオフセットベクトルを生成する。

【0032】コマンド状態マシン 200でコマンドコード

を受取り、ハードウェア制御信号を生成する。そのコマンドは次に、動作待ち行列 230に対して出力する前にコマンドコードをオフセットベクトルに変換するジャンプ表を含む一次コマンド/アドレス/データ待ち行列 210に送る。生成されるハードウェア信号の種類はシステムの構成による。例えば本実施例では、ハードウェア信号を生成して、ページバッファが使用されるのか、どのページバッファが使用されるのか、行われる動作は8ビット動作あるいは16ビット動作なのかなどを識別する。

【0033】図4～図14に戻る。いくつかのコマンドコードとアレイコントローラオフセットベクトルはアレイコントローラ50が行うカスタムアルゴリズム用に割り当てられている。フラッシュメモリシステム 310をカスタム化するには、アレイコントローラプログラムメモリに単にコードをロードしてカスタムアルゴリズムを行うだけである。従ってフラッシュメモリシステム 310は特定のユーザのアプリケーション専用のコマンドを行うようにカスタム化できる。更にコマンドセットは柔軟性があり、ユーザインターフェイス回路40を変更することなしに新しいコマンドを加えたり、古いコマンド機能を変更できる。

【0034】コマンド機能はプログラムメモリ内の最初の32位置に格納されたプログラムメモリアドレスを改訂して異なるプログラムメモリ位置を引用することで容易に変更することができる。更にアレイコントローラ50の機能は製造時にシステムに与えられたアルゴリズムに限定されない。システムは製造後にアレイコントローラアルゴリズムの更新、修正、追加を可能にするフラッシュプログラムメモリなどの更新可能プログラムメモリを備えている。好適にはプログラムメモリはCPU（ユーザ）が出す所定のコマンドを通して更新する。ユーザインターフェイスはコマンドを受けると、アレイコントローラがプログラムメモリを更新できるようにするページバッファに一時格納されたアルゴリズムをアレイコントローラに実行させる。このようにしてアルゴリズムの開始アドレス及びアルゴリズムそれ自身を修正、削除、あるいは追加できる。例えばプログラムメモリはユーザから受け取る新しいアルゴリズムで更新することができる。

【0035】アレイコントローラ50によりコマンドを実行する場合、コマンド、アドレス、データ情報をユーザインターフェイス回路40で処理するため一時待ち行列 210に与える。特に一時待ち行列 210でコマンド、データ、アドレス情報を受け取りコマンドがオフセットベクトルに変換されると、情報は動作待ち行列 220に送られる。アレイコントローラコマンドがコマンド状態マシン 200に与えられると、コマンド状態マシン 200はそのコマンドとアドレス/データ情報を一時待ち行列 210に送り、それはそれらを引続き動作待ち行列 230に送る。一時待ち行列 210は動作待ち行列 230が活動待ち行列の1

つにそのコマンドを受け入れる用意ができるまでそのコマンドを保持する。一時待ち行列 210への情報の転送は、ユーザ制御バス 106を通して受け取る書込み可能クロックと同期化する。動作待ち行列 230はコマンドを一時待ち行列 210から受取り、それをアレイコントローラクロックで駆動される活動待ち行列に配列する。コマンド状態マシン 200が情報を一時待ち行列 210に転送すると、それは更にコマンドが活動待ち行列に加えられるのを待っていることを動作待ち行列 230に告げるフラッグを設定する。動作待ち行列 230がコマンドを一時待ち行列 210から待ち行列の1つに移動すると、フラッシュはリセットされる。このフラッグは更に状態レジスタ 260により待ち行列満杯ビットとして用いられる。この待ち行列満杯ビットはそのビットがなくなるまでアレイコントローラ50により実行することが必要なコマンドをフラッシュメモリシステム 310に出さないようにユーザに告げるのに用いる。

【0036】好適には動作待ち行列 220は2動作迄待ち合わせさせる能力を持っているが、2動作以上待ち合わせさせることができることは明かである。一次待ち行列は実行する動作ないし実行されている動作を示す。二次待ち行列は一次待ち行列内にある動作の実行が完了すれば行われる次の動作のデータを含んでいる。アレイコントローラ50による一次待ち行列内の動作の実行が完了すると、一次動作はその動作待ち行列から取り除かれてアレイコントローラ50により次の動作を実行できるようにする。実行する命令が一時待ち行列 210に格納されている場合、オフセットベクトル、データ、アドレスは引き続き実行するために動作待ち行列 230に転送される。後に詳細に説明するように、特定のケースでは一次動作が完了する前に二次待ち行列内にある動作を行うことが望ましい。一次及び二次動作の間で革新的なコンテキスト切替え処理を行って一次待ち行列にある動作の完了前に二次待ち行列内にある動作を実行する。

【0037】ユーザインターフェイス回路40はアレイコントローラ50により実行するコマンドを受け取ると、アレイコントローラ50を駆動するクロック信号を出すアレイコントローラ50の局所発振器が開始するようにアレイコントローラ50に信号を出す。アレイコントローラ50が作動すると、オフセットベクトルはユーザインターフェイスからアレイコントローラ50に転送され、プログラムメモリを割り出す。動作待ち行列 230に格納されたアドレス、データ情報は直接読取り/書込み経路回路30に与えたり、制御レジスタから与えることができ、読取り/書込み経路回路30でのアドレス、データ情報を用いての動作の実行は、プログラムメモリから実行されるアルゴリズムに規定されてアレイコントローラ50により制御される。

【0038】動作待ち行列 220は好適には、一時待ち行列 210からオフセットベクトルとアドレス、データ情報

を受取り、アレイコントローラ50によりアクセスするベクトル及び関連アドレス、データ情報を待ち合わせさせる状態マシンとなる。

【0039】実施例では、コマンドの待ち行列とユーザインターフェイス回路40へのパイプライン化を支援するため、3層のコマンドの待ち行列を造る。例えば受け取ったコマンドがアレイコントローラ50の動作を必要とし、1つが現在実行しているものでない場合、ユーザインターフェイス回路40は動作待ち行列の最初のもの（一次待ち行列とも称する）を動作のためのデータと共にロードしてアレイコントローラ50の作動を開始する。一般にアレイコントローラ50はその自由裁量で、そのコマンド情報の内容を待ち行列の最初に持ち、行列の最後はユーザバス 306を通しコマンド状態マシン 200そしておそらく一時待ち行列 210を通して出される別のコマンドに使用できるようにする。このアーキテクチャによりアレイコントローラ50が最初のコマンドを実行している間に、ユーザはアレイコントローラ動作その他の任意の有効なコマンドをユーザインターフェイス回路40に出すことができる。ユーザインターフェイス回路40は後続のコマンドを受け取ると、新しい動作が待ち行列に配列されたことをアレイコントローラ50に通知し、現在走っているアレイコントローラアルゴリズムはアレイコントローラ50がその動作を中断して待っている動作を扱うべきかあるいは処理中の動作を最初に完了するかを判定する。アレイコントローラ50が現在実行しているアルゴリズムを中断するかどうかを判定する基準は、実行しているアルゴリズムの論理に含めるようにする。例えばブロックを消去するアルゴリズムは、後続のプログラムコマンドは消去アルゴリズムの実行を中断することができることを示すコードを含むようにする。

【0040】ユーザインターフェイス回路40の構造により、動作待ち行列 230内でコマンドをパイプライン化できる。例えば1プログラムバイト/語ないし2プログラムバイト/語コマンドをパイプライン化でき、1プログラムバイト/語消去、単一ブロックをパイプライン化でき（メモリアレイの異なるブロックに関して）、ページバッファコマンドからの1つのブロックの消去と別のブロックの書き込みをパイプライン化でき、ページバッファコマンドからのプログラムレイと別のブロックの消去並びに他の特殊なアルゴリズムをパイプライン化できる。

【0041】ユーザインターフェイス回路40には複数の状態レジスタ 260が含まれている。状態レジスタの一部はアレイコントローラ50による読取り、書き込みアクセスが可能で、ユーザインターフェイス回路40による読取りアクセス可能であるが、他のレジスタはユーザインターフェイス回路40による読取り、書き込みアクセス可能である。各々の状態レジスタは、アレイコントローラ動作の実行状態に関する情報をアレイコントローラによりモニ

タないし交信でき、ユーザインターフェイスとユーザがモニタできるようにするため、ユーザインターフェイスにより読取りアクセス可能となっている。ユーザはコマンドを出していつでも状態レジスタにアクセスして、アレイコントローラの状態を判定することができる。例えばメモリスシステムにコマンドを出す前に、アレイコントローラの状態が出す命令が有効なコマンドとして受け入れられる状態であることが好ましい。アレイコントローラの状態は状態レジスタ 260を読み取ることで判定することができる。

【0042】好適には3つの大域状態レジスタと1組の32ブロック状態レジスタ（BSR）を設ける。図15は例示的な状態レジスタビット定義を示すブロック図である。大域状態レジスタは装置の全般的な状態を提供するが、ブロックの特定情報は提供しない。例えば間違った動作は大域状態レジスタで検出することができるが、間違った動作が生じた実際のブロックはブロック状態レジスタを読み取ることでしか検出できない。第1の大域状態レジスタは先のメモリ生成物と協調する反対方向に構成するようにする。第2の大域状態レジスタはアレイコントローラの状態、動作待ち行列の状態、ページバッファの状態に関する情報を提供する。第3の大域状態レジスタはアレイコントローラにより操作できる8つの非専用ビットを含んでいる。そのビットの定義は、アレイコントローラ上で実施されるアルゴリズムによる。従ってユーザはコマンドを出して第3の大域状態レジスタをモニタして特定のアルゴリズムに対するアレイコントローラにより実行状態を判定することができる。状態レジスタを通してのアレイコントローラの状態をモニタする機能により、ユーザはメモリスシステムに引続きコマンドを出す前に有効なコマンドを判定することができる。

【0043】ブロック状態レジスタ（BSR）はアレイのブロックに対して行われる動作の状態を維持するのに用いる。図16に例示的なブロック図を示す。図16に示すように、ブロック状態レジスタは32x8単一ポート書き込み二重ポート読取りSRAMアレイとして構成するのが好ましい。SRAMの書き込みポートはアレイコントローラ書き込みデコーダに接続されている。読取りデコーダはアレイコントローラとユーザインターフェイスに備えられている。従ってアレイコントローラはSRAMアレイにあるブロック状態レジスタに対して書き込み、読取りを行うことができるが、ユーザインターフェイスはアレイから読取りしか行うことができない。ユーザは、1つのブロック状態レジスタを選択し、入出力状態レジスタバス上のブロック状態レジスタの内容をユーザインターフェイスに与える信号（例えばアドレスを有するIDA [12:0]、IDA [20:16] 及びOEBPAD）をもたらし要求をユーザインターフェイスに対して出すことによりブロック状態レジスタを読み取ることができる。そこで状態情報がユーザにもたらされる。図17はブロッ

ク状態レジスタがSRAMアレイで構成されるときブロック状態レジスタのビットの例示的な構成を示すもので、ビットの定義はプログラム可能でシステムの作動に更に柔軟性をもたらす。

【0044】各々のブロック状態レジスタはフラッシュアレイの対応するブロックの書き込み保護のメカニズムをもたらすロックビットを含み、不注意な消去やプログラム動作からブロックを保護するため、その特定ブロックと関連したブロック状態レジスタ内の不揮発性ロックビットを設定する。特定ブロックに対してロックビットを設定すると、そのブロックは消去ないし書き込み動作から保護されるようになる。ロックビットは各々のアレイブロックに有り、関連ブロックが消去される度に消去される。ユーザはロックビットの状態をブロック状態レジスタ内の対応するビットを読み取ることにより読み取ることができる。

【0045】上述したように、アレイコントローラはフラッシュ装置の内部モードを制御するために使用するプログラマブル・マイクロコントローラである。特にアレイコントローラはフラッシュアレイ内のセルをプログラム、消去するアルゴリズムを含むアルゴリズムを自動的かつ正確に制御する手段を提供する。アレイコントローラはアレイコントローラ動作の起動を制御するユーザインターフェイスを通してアクセスする。アレイコントローラは動作当り2クロックサイクルのベースで作動する。好適には動作Nの実行の用意に対するクロックサイクルは、動作N-1の実行クロックサイクル中に生じる。このようにしてアレイ動作は1サイクルで実行し、第1の命令の実行前に遅延だけが生じる。クロックサイクル内で命令実行は3つの段階に分ける。この区分により1クロックサイクル中に入出力バスサイクルのための十分なクロックエッジがもたらされる。

【0046】図18で、アレイコントローラはフラッシュプログラムメモリ320に記憶されたアルゴリズムを実行する。それらのアルゴリズムは命令語、特にデータ転送命令、計算命令、分岐命令、制御命令からなる。データ転送命令は8ビットないし16ビットデータをレジスタファイルに対して移動することに関連している。分岐命令はサブルーチンコールや条件的／無条件的ジャンプの使用を通してプログラミングでアルゴリズムの流れを修正可能にする。計算命令は論理演算装置370に関連した動作をもたらす。制御命令はフラッグを設定ないし除去し、割込み処理ルーチンに対するポインタを設定する手段を提供する。

【0047】ユーザインターフェイスとアレイコントローラの間にある制御アクセス回路380は、アレイコントローラが命令を受けてアルゴリズムを実行し、状態情報をユーザインターフェイス及び最終的にユーザに通信する手段を提供する。例えば実施例では、ユーザインターフェイスは実行を待っているアルゴリズムがあることを

アレイコントローラに知らせるCDRUNF、現在アルゴリズムに加えて実行を待っている少なくとも1つのアルゴリズムがあることを示すCDCMDRDY、アレイコントローラに未決のアレイコントローラ実行を一時停止する要求があることを知らせるCDSUSREQの3つの信号を出す。代わりにアレイコントローラはユーザインターフェイスに対してアレイコントローラが装置の制御を有して走っているかどうかを示す信号FDRDY、アレイコントローラが一次待ち行列にあるコマンドないし二次待ち行列にある割込みコマンドを実行しているかを示すFDNXTCMD、現在行っている動作が完了したことを示すFDOPDONE、動作はうまく一時的に停止し、ユーザはメモリ装置の制御をして読取り動作などの動作を行うことができることを示すFDIDLEを通信する。

【0048】先に述べたように、ユーザインターフェイスはマイクロプロセッサバス信号により駆動されるが、アレイコントローラはフラッシュシステム内にある発振器により駆動される。特に発振器位相生成器ブロック305はアレイコントローラ回路のために離散クロック信号として用いられる3つの非重複クロックパルスを生成する。発振器305はユーザインターフェイスからのコマンドが実行されるときにユーザインターフェイスにより起動される。

【0049】プログラムメモリ320はアクセスしたアルゴリズムをユーザインターフェイスから受け取ったジャンプベクトルとプログラムメモリの最初の32アドレス内に格納されたプログラムアドレスにしたがって格納し、アレイコントローラにより実行して要求された機能を行うアルゴリズムを格納する。プログラマブルアレイコントローラを備えることにより実現される利益は膨大である。アルゴリズムはフラッシュシステムが提供するアプリケーションの要件に対して構成することができる。例えば標準プログラム及びプログラム及び消去過程上で変更を行う消去過程並びにカスタム過程のアルゴリズムはプログラムメモリに含めて対応するコマンドを出してアクセスすることができる。更にユーザインターフェイスに対して出したコマンドを通してユーザはプログラムメモリ内に格納されたアルゴリズムを修正することでアレイコントローラの機能を修正することができる。これはアルゴリズムを変更するにはハードウェアを修正する必要のあった従来技術の装置とは全く異なる。好適にはプログラムメモリはユーザインターフェイス回路により制御されるようにアレイコントローラにより修正されるようにする。プログラムメモリ自身が修正されると、ページバッファにはプログラミングアルゴリズムがロードされ、アレイコントローラはプログラムメモリをプログラムするためアルゴリズムに付いてページバッファを参照する。

【0050】本発明のフラッシュメモリシステムのアー

キテクチャによりもたらされる柔軟性は非常に広い。例えばアプリケーションがフラッシュアレイ内に格納された特定のデータを指定するためにインデックスを備えていることを必要とする場合、従来のシステムはSRAMなどの外部メモリを利用してフラッシュアレイ内のどこかのデータにアクセスするかを判定するためにポインタを格納してアクセスしていた。しかし本発明のフラッシュシステムにより、アレイコントローラはポインタを格納してそのポインタを用いて探索を行うことができ、外部SRAMの必要性をなくし、各々のアレイは動作を並列に行うことができるので動作のスループットを増大することができる。従ってアレイのためのフラッシュの並列処理が達成される。本発明のフラッシュメモリシステムにより実行できる別の応用は、フラッシュメモリを用いてカメラのフィルムと置き換えることである。この応用を効果的なものにするため、画像データの高速書き込みが必要になる。一般にエラーが引き起こされると、別の書き込み動作の試みが行われる。しかしこの応用では速度が不可欠であり、単にそのブロックを不良としてマークする方が早い。本発明のフラッシュメモリシステムは単にプログラムコードを生成して書き込み機能を行い、そのコードをフラッシュアレイコード記憶装置に格納することでこの応用に対応することができる。このようにカスタム化が可能で、実施は非常に簡単である。

【0051】アルゴリズムを実行するため、アレイコントローラは現在命令を記憶する命令レジスタ 330、プログラムカウンタ 340、コールスタック 345、論理演算装置 (ALU) 370、実行中にALU 370によりスクラッチメモリとして使用されるレジスタファイル360からなる汎用処理構造として具現される。

【0052】プログラムカウンタ 340はアレイコントローラに与えられる命令の適切なサイクルを維持するために全てのポインタ記憶装置と論理を含んでいる。更にこの構造は革新的なコンテキスト切替え、割込み構造を備えている。プログラムカウンタの基本的な動作は、実行する次の命令を解読し、次の適切な命令を指定することである (EDPC [11:0])。プログラムカウンタの出力として与えられるアドレスを用いてプログラムメモリから次の命令を検索し、命令レジスタに転送してALUにより実行する。図19で、プログラムカウンタはコール 805、コール割込み 810の2つのスタックと、割込み開始 815、割込み戻し 820、ゼロ充填 825、オフセットジャンプ 830、間接ジャンプ 835の5つのレジスタと1つのレジスタ加算器 840からなる。制御回路 845は、マルチプレクサ 855の制御を含む複数の制御機能を行い、835、815、820、825、830、840、805の様々なソースからアドレス出力のソースを選択し、次の命令ジャンプ位置アルゴリズム、オフセット開始点、ハードウェア割込み、ソフトウェア戻しの間の仲裁を行う。コールスタック 805は12ビットワイドのプッシュアップ・ポップダウン

スタックからなる。

【0053】このアーキテクチャによりアレイコントローラは割込み中にコンテキスト切替えを行って、割込み動作が完了した後コンテキスト切替えを行ってアレイコントローラの元の状態を回復し、元の動作を続行できるようにタイミングよくアレイコントローラのコンテキストを保存することができる。各々のスタック 805、810は、スタックのような挙動をもたらすためロジック (図示せず) で囲んだラッチからなる。好適にはプッシュ、ポップロジックはプッシュないしポップ動作が行われた後指定する次のアドレスを解読する状態マシンとする。

【0054】図18に戻り、プログラムメモリ 320はプログラムカウンタ 340により決定されたアドレスを受け取り、命令出力をラッチする命令レジスタ 350に命令を出力する。動作で更に柔軟性をもたらすため、命令レジスタ 330は更にページバッファ 310を通して受け取る命令をラッチする機能を持っている。その命令は次にALU 370により実行する。ALU 370は全ての演算、論理機能並びにアレイコントローラのためのシフト動作を行い、またバイトと語動作の両方を扱うことができる。実行する命令は命令レジスタ 350を通してプログラムメモリ 320から得る。ALUへの入力データは、ページバッファ及び入出力インターフェイスを初めとする複数のソースから発することができる。

【0055】レジスタファイル 360は3つのポートのSRAMとして構成され、アレイコントローラが2つのポートから読取り、第3のポートを通してレジスタファイルに書き込むことができるようにしている。2つの読取りポートと1つの書き込みポートの各々は互いに独立して作動し、実際には同一メモリ位置で作動することができる。そのタイミングはセルの段階1中に読取り動作が生じ、クロックの段階3中に書き込み動作が生じ、それにより段階2を計算段階として使用するというものである。従ってレジスタファイルは1クロックサイクルで読取り、更新することができる。レジスタファイルはアレイコントローラの状態を含み、特にアレイコントローラアルゴリズムの実行中に利用する変数を含んでいる。

【0056】レジスタファイル 360は2つの部分に区分されている。主要部分である第1の部分はアレイコントローラが実行するアルゴリズムの変数を含み、第2の部分は割込みアルゴリズムの変数を含んでいる従って割込みを行うとき、レジスタファイルは主要部分から第2の部分へのアレイコントローラの状態のハードウェアコンテキスト切替えを行うことができる。ここで全ての動作は主要部分と同一に見えるレジスタファイルの割込み部分で行われる。レジスタファイルの主要部分に保持された全ての変数は維持されるが、割込み作業中にアクセスすることはできない。割込み作業後、制御は主要部分に再び与えられる。

【0057】本アーキテクチャは、4レベルの深さのプ

ログラムカウンタスタックを支援する。これによりメモリ装置に対して書き込まれるアルゴリズムは動作をよりモジュール的にすることができる。例えば戻り命令はプログラムカウンタスタックと共に二重のデューティを行う。プログラムカウンタスタックにデータがあるとき、戻り命令は被呼サブルーチンから戻す役割をする。PCスタックが空の場合、この命令はアルゴリズム終了命令の役割をする。これにより通常動作に付いてはアレイコントローラが実行を停止するがサイクリングなどのテスト動作に付いてはユーザアルゴリズムをそれらがサブルーチンであるかのように呼び出すことができるようにユーザアルゴリズムを戻り命令で終了することができる。

【0058】本発明のフラッシュメモリシステムのアーキテクチャは、データの変改ないし高電圧プログラム/消去回路なしにアレイコントローラアルゴリズムの割込みを可能にする革新的な割込みメカニズムを提供する。従来の装置とは異なり、この割込みメカニズムはアレイに対して害を与える恐れのある状態からメモリ装置を取り出すことで割込みを行う前にメモリ装置を保護する方法を提供する。例えばアレイコントローラが現在消去過程を制御している場合、アレイを消去するのに必要な電圧はメモリに害を与えるのを避けるために低下させる。割込みを行った後、メモリ装置とアレイコントローラを消去パルスのような元の実行過程を続行できる状態に配置し直す。

【0059】特に革新的な割込みメカニズムは最初に現在実行しているアルゴリズムに関連した割込み開始コードを実行することで現在実行しているアルゴリズムの安全な割込みと中断を備えている。この割込み開始コードは現在実行アルゴリズムを安全に中断する。例えば割込み開始コードは利用している高電圧回路を低下し、読取り/書き込み回路を割込みアルゴリズムで使用する状態に置き、必要なレジスタのコンテキスト切替えを行って中断するアルゴリズムの安全時点を判定する。割込み開始ルーチンが実行を完了すると、割込みルーチンが実行を始め、割込みルーチンの実行の終わりに、割込み戻しルーチンが実行されてプログラムカウンタを調節して割り込んだコード内の安全戻し地点を指定し、被割込みルーチンが実行を続行できるようにレジスタのコンテキスト切替えを行う。

【0060】本発明のフラッシュ回路に備えられた革新的な割込み処理を用いて、消去中のプログラム支援アルゴリズムその他のアルゴリズム中断機能が可能となる。例えばこの割込み構造により、アルゴリズムを同時に実行している間に割込みを生じることが可能である。更に割込み構造、動作待ち行列及び状態レジスタは革新的な仮想状態マシンとして機能してマルチプロセッシング的な機能を提供する。

【0061】割込みの取り扱いはユーザインターフェイスの状態マシンが出す3つの命令、即ち使用可能割込み

(ENI)、使用不能割込み(DSI)、セット割込み開始レジスタ(SISR)により制御する。ENIはアルゴリズムへの安全な戻し地点を引数として取り、割込み戻しレジスタ(IRR)をこの値に設定し、割込み可能フラグ(IF)を設定する。DSIは引数を取らず、単にIFをリセットするだけである。SISRは割込み開始ルーチンアドレスのアドレスを引数として取り、割込み開始レジスタ(ISR)を設定する。PCスタックの保全性を確保するため、IRRアドレスは割込みが為された手順内になければならない。従って例えば信号CDCMDRDYが実行を待っているコマンドを示し現在実行しているアルゴリズムが割込み可能である場合、即ちIFビットが設定されている場合、CDCMDRDYが設定された後最初の命令サイクルでISRに格納されているアドレスへのジャンプが生じる。直ちに割込み要求信号が表明され、IFが取り除かれる。割込み開始ルーチンが完了すると、戻しによる表示(RET)命令、割込み肯定応答信号及び開始信号が表明される。そこで割込みアルゴリズムが実行される。空のPCスタック上でRETにより表示される割込みアルゴリズムが完了すると、割込みビットが取り除かれ、IRRに記憶されたアドレスへのジャンプが生じる。そこで被割込みアルゴリズムは実行を続ける。

【0062】戻しアドレス及び割込み開始ルーチンを含む割込み戻しルーチンは事前に決定されている。アレイコントローラアルゴリズムが造られプログラムメモリにロードされるとき、割込み開始ルーチンと戻しルーチンが造られてプログラムメモリにロードされ、割込み開始ルーチンと戻しルーチンのアドレスを決定するようにする。アルゴリズムが実行されるとき、割込みが生じたときに情報を容易に得ることができ、割込み開始及び戻しルーチンの開始アドレスをタイミリーに利用してプログラムメモリの該当アドレスにアクセスして命令レジスタにロードしALUにより実行するようにIRRとISRはプログラムカウンタにロードされる。

【0063】このように割込み可能アルゴリズムが実行しており、割込みが生じると、制御は直ちにそのアルゴリズムの割込み開始ルーチンに渡される。割込み開始ルーチンはアレイコントローラを割込みアルゴリズムが実行し被割込みアルゴリズムが続いて実行されて完了できる状態にする。従って割込み開始ルーチンは割込みをこの時点で行うことができるかどうかを判定し、そしてそれを行うことができればフラッシュ高電圧と読取り回路を周知の安全な状態にする。そこでアレイコントローラはコールスタックとレジスタファイルのコンテキスト切替えを行って交互のコピーにアクセスし、それによりコンテキスト切替えを行う。割込みアルゴリズムを完了すると、制御は割込み戻しルーチンに渡される。この移行が行われると、コールスタックとレジスタファイルは被割込みアルゴリズムで使用されたものとものに切り換え

られる。割込み戻しルーチンは装置の高電圧と読取り回路を適切な状態にして被割込みアルゴリズムの実行を続け、実行を始める安全地点である被割込みアルゴリズムのアドレスを提供することに責任を持つ。この時点で被割込みアルゴリズムの実行が続けられる。この過程の流れを図20に図式的に示す。

【0064】被割込みアルゴリズムの実行が始まる地点はアルゴリズムが割り込まれた場所とは同じでないことがあることに留意する。これはアプリケーションにより決定される。例えば被割込みアルゴリズムがアレイ消去動作ならば、割込み前に行われた最後の消去動作が完了したことを検証するためにアルゴリズムは割込み後、検証命令で開始されることがある。アルゴリズムのアプリケーションが指令するように被割込みアルゴリズムの戻し地点を決定する他の基準を用いることができる。

【0065】更に被割込みアルゴリズムの安全戻し地点を識別するアドレスは、アルゴリズムがアルゴリズムの異なる「段階」ないし部分を実行することでアルゴリズムの実行中に変わることがある。従って割込み戻し地点はコードの実行中に更新されて異なる戻し地点を指定する。これは割込み戻しルーチンを修正ないしIRRを更新して、アルゴリズム内の実行位置により異なる割込み戻しルーチンを指定することで行うことができる。例えば消去アルゴリズムでは、事前調整段階の戻し地点は、アルゴリズムの消去部分の戻し地点とは異なり、それはまたアルゴリズムの後調整段階の戻し地点とは異なるものとなる。

【0066】好適には割込みフラッグと大域割込みフラッグの2つの割込みフラッグをアレイコントローラに設ける。割込みが生じるには、動作が未決で、両フラッグが使用可能でなければならない。最初は使用不能状態の割込みフラッグは、アルゴリズム内で使用して局所的に割込みを可能、不能にする。従ってアルゴリズムは、アルゴリズムの実行時点でそれぞれ割込み可能及び割込み不能であるフラッグを使用可能、不能にするアレイコントローラに対する命令を持つことができる。割込みフラッグはコールないし戻し命令が実行されると自動的に不能にされる。これは特定サブルーチンに付いてIRRが適切に戻しを指定するようにするためである。

【0067】最初は使用可能になっている大域割込みフラッグは、アルゴリズムが未決割込みを扱うことができない場合に割込みを不能にするのに使用する。図21に示すように、被割込みアルゴリズムに対する割込み開始ルーチンがこの時点で割込みを処理することができないと判定すると、大域割込みフラッグが不能になり、割込み戻しルーチンに対するスキップ命令が実行される。

【0068】スキップ命令は依然割込み開始ルーチン中に実行されることを除いて戻し命令と見ることができ。スキップ命令によりアレイコントローラは割込み開始ルーチンを終了し、直ちに割込み戻しルーチンを実行

して割込みアルゴリズムの実行をスキップする。従って割込み戻しルーチンが完了した時点で大域割込みフラッグが使用可能であれば、割込みアルゴリズムが完了し、動作待ち行列から取り除くことができる。大域割込みフラッグが不能であれば、ユーザインターフェイスは未決の動作を続いて実行できるように動作待ち行列に残す。

【0069】従ってこの構造はアレイ内のデータの変改や破壊を生じることなく事実条件的にフラッシュメモリシステム内で作動する割込みメカニズムを提供する。更にこのメカニズムは割込みの優先順位レベルに容易に対応することができる。例えば各々のアルゴリズムは優先順位レベルで識別できる。ユーザインターフェイス回路40は動作を二次待ち行列にロードし、割込みを要求するのに必要な信号を出し、現在実行している過程は未決のコマンド要求は高い優先度のものかどうかを判定する。そうであればその割込みは実行している動作を中断してアレイコントローラにより実行される（一次待ち行列に現在格納されている情報で記されているように）。

【0070】更にこの構造により消去動作中のプログラムが生じる。アレイ上の消去動作はよく知られた時間のかかる過程である。ブロックの消去動作が現在実行しており、別のブロックのプログラム動作が実行を待っている場合、割込みが生成され割込みアルゴリズムの適切性を割込み開始ルーチンがチェックし（例えばそれは既に消去しているブロックへのプログラムしようとする試みか？）、プログラムアルゴリズムの実行前に内部電源をオフ状態に安全にシーケンス化する。これを図22、23に単純化した流れ図で例示する。説明のために消去過程は全てのブロックデータが「0」にプログラムされた事前調整過程とブロックデータが「1」の値に設定された検証過程の2つの大きな部分からなると記述できる。過程流れ図に示すように、例えば別のブロックをプログラムする要求により生じる割込みは、高電圧回路操作中には許されないが、消去中には許される（図23を参照のこと）。

【図面の簡単な説明】

【図1】 フラッシュメモリを含むコンピュータシステムのブロック図である。

【図2】 本発明のフラッシュメモリシステムのブロック図である。

【図3】 本発明のユーザインターフェイスのブロック図である。

【図4】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図5】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図6】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表

である。

【図 7】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 8】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 9】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 10】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 11】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 12】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 13】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 14】 本発明のシステムで受け取るコマンドコードと実施されるアレイコントローラベクトルを例示した表である。

【図 15】 例示的な状態レジスタビット定義を示したブロック図である。

【図 16】 本発明のブロック状態レジスタの構造を示したブロック図である。

【図 17】 ブロック状態レジスタのビットの例示的な構成を示す。

【図 18】 本発明の不揮発性メモリで利用するアレイコントローラのブロック図である。

【図 19】 プログラムカウンタ回路を示すブロック図である。

【図 20】 本発明の割込み過程の流れを示す図である。

【図 21】 本発明の割込み過程の流れを示す図である。

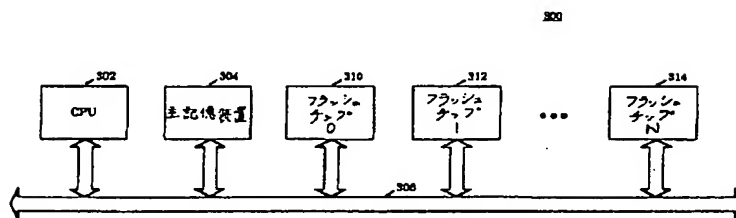
【図 22】 本発明の教示による消去過程中的プログラムを示す流れ図である。

【図 23】 本発明の教示による消去過程中的プログラムを示す流れ図である。

【符号の説明】

- 300 コンピュータシステム
- 302 CPU
- 304 主記憶装置
- 306 ユーザバス
- 310 フラッシュメモリシステム
- 20 フラッシュセルアレイ
- 30 読取り／書込み経路回路
- 40 ユーザインターフェイス回路
- 50 アレイコントローラ
- 60 制御アクセス回路
- 70 ページバッファ回路
- 80-85 制御レジスタ回路
- 200 コマンド状態マシン
- 210 一時待ち行列
- 230 動作待ち行列
- 250 ページバッファカウンタ
- 260 状態レジスタ

【図 1】

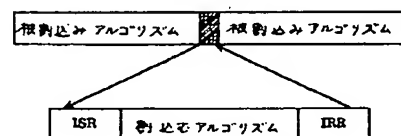


【図 10】

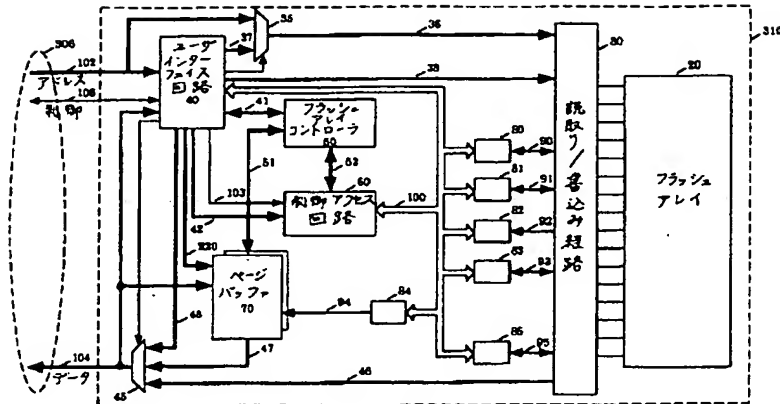
コマンド コード	フェイス ベクトル	フェイス動作
0 A	0 0	このフェイスベクトルは3サイクルページバッファ付き書込みコマンド、すなわち「ページバッファからバイト／語をプログラム」コマンド（コード 0 C H）用に予約されている
1 A	0 1	カスタムアルゴリズム

ページバッファなし3サイクル書込み

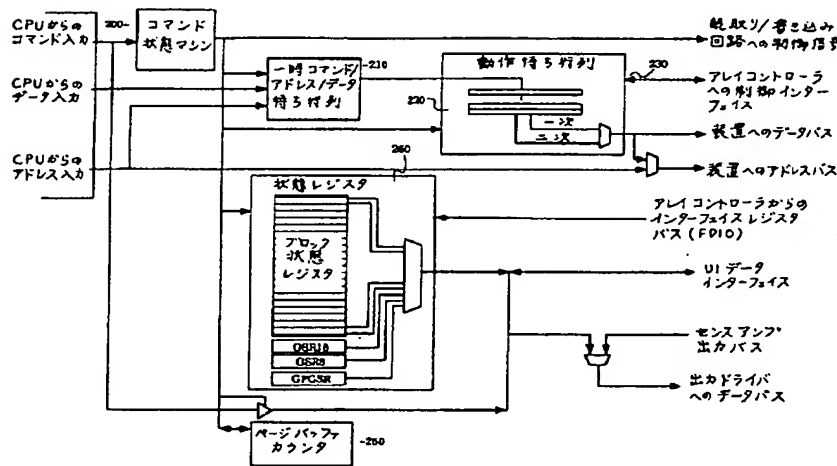
【図 20】



【図2】



【図3】



【図6】

【図7】

コマンドコード	フェイスベクトル	フェイス動作
07	1D	カスタムアルゴリズム
87	1E	カスタムアルゴリズム
F7, 06又は07	00	このフェイスベクトルは3サイクルページバッファ付き書き込みコマンド、すなわち「ページバッファからバイト/ワードをプログラム」コマンド(コード0C11)用に予約されている。

ページバッファなしの3サイクル書き込み

コマンドコード	フェイスベクトル	フェイス動作
79	17	このフェイスベクトルは3サイクルページバッファなし書き込みコマンド、すなわち「単一ブロック消去」(コード20H)用に予約されている。
18	01	カスタムアルゴリズム
28	02	このフェイスベクトルは3サイクルページバッファなし書き込みコマンド、すなわち「フェイスアルゴリズムでテストモードを使用可能にする」コマンド(コード2AH)用に予約されている。
38	03	ページバッファにアルゴリズムコード改訂1Dをダンプ。データD0は確認として使用する。データを選択したページバッファのページバッファ位置0に入れる。
48	04	カスタムアルゴリズム
58	05	カスタムアルゴリズム
68	06	カスタムアルゴリズム
78	07	カスタムアルゴリズム
88	08	カスタムアルゴリズム

ページバッファ付き32サイクル書き込み

【図4】

コマンド コード	フェイス ベクトル	フェイス動作
20	17	単一ブロック消去。10は2番目のサイクル中に与えられるアドレスにより指定されたブロックを消去する設定。2番目のサイクルはこのコマンドを選択的に使用する（確認）
16	01	カスタムアルゴリズム
26	02	このフェイスベクトルは3サイクルページバッファなし書込みコマンド、すなわち「フェイスアルゴリズムでテストモードを使用可能にする」コマンド（コード3AH）用に予約されている。
36	03	このフェイスベクトルは3サイクルページバッファ付き書込みコマンド、すなわち「ページバッファにアルゴリズムコード改訂1Dをダンプ」用に予約されている。
46	04	カスタムアルゴリズム
56	05	カスタムアルゴリズム
66	06	カスタムアルゴリズム
76	07	カスタムアルゴリズム
86	08	カスタムアルゴリズム
96	09	レディ/使用中ピンアキタクチャの再構成。チップは非パルス化オーブンドレイン・レディ/使用中パルファに対してデフォルトする。このコマンドを出した後、レディ/使用中出力は「パルス化」レディ/使用中方式に交換される。
A6	0A	カスタムアルゴリズム
B6	0B	カスタムアルゴリズム
C6	0C	カスタムアルゴリズム
D6	0D	カスタムアルゴリズム
E6	0E	カスタムアルゴリズム
F6	0F	カスタムアルゴリズム

2サイクルページバッファなし書込み

【図5】

コマンド コード	フェイス ベクトル	フェイス動作
40	1F	バイト/語。プログラムバイト/語シーケンス。コマンド「10」もこの「10」コマンドに写されるが、ちょうど同じものではない（AMD互換性コマンド）
17	11	カスタムアルゴリズム
27	12	カスタムアルゴリズム
37	13	カスタムアルゴリズム
47	14	カスタムアルゴリズム
57	15	カスタムアルゴリズム
67	16	カスタムアルゴリズム
77	10	NVロックビット設定。WPB PADピンの状態によりNVロックビットを設定（WPB PADはコマンドが実行できるように書くなければならない）。2番目のサイクルでは、データは確認コードとして使用されるD0Hでなければならない。アドレスはロックするブロックを選択する
87	18	カスタムアルゴリズム
97	19	NVロックビットをロード。このアルゴリズムはアレイ消去ブロックの各々のNVロックビットの状態をブロック状態レジスタのブロックロック化ビット（ビット6）にロードする。2番目のサイクルで、データは確認コードとして用いられるD0Hでなければならないが、アドレスは無定義。
A7	1A	全チップ消去。A7はNVロックビットで保護されていないチップの全ブロックを消去する設定である。2番目のサイクルはこのコマンドをうまく用いるためにそのデータにD0を持たなければならない（確認）
B7	1B	全ブロックをロック
C7	1C	カスタムアルゴリズム

ページバッファなしの2サイクル書込み

【図12】

コマンド コード	フェイス ベクトル	フェイス動作
20	12	カスタムアルゴリズム
30	13	カスタムアルゴリズム
40	14	カスタムアルゴリズム
50	15	カスタムアルゴリズム
60	16	カスタムアルゴリズム
F0	1F	X8 8ビットで語をプログラム。（時間を節約するため）2つのバイトプログラムコマンドを行わずにX8 8ビットに2バイトのデータをプログラムするのに使用する
80	18	カスタムアルゴリズム
90	19	このフェイスベクトルは2サイクルページバッファなし書込みコマンド、すなわち「NVロックビットをロード」（コード17H）用に予約されている
A0	1A	このフェイスベクトルは2サイクルページバッファなし書込みコマンド、すなわち「全チップ消去」（コードA7H）用に予約されている
70	17	このフェイスベクトルは2サイクルページバッファなし書込みコマンド、すなわち「単一ブロック消去」（コード10H）用に予約されている
B0	1B	カスタムアルゴリズム
C0	1C	カスタムアルゴリズム
D0	1D	カスタムアルゴリズム
E0	1E	カスタムアルゴリズム

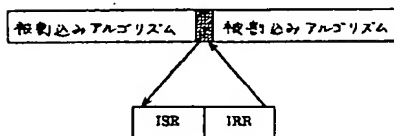
ページバッファなし3サイクル書込み

【図9】

コマンド コード	フェイス ベクトル	フェイス動作
B9	1B	カスタムアルゴリズム
C9	1C	カスタムアルゴリズム
D9	1D	カスタムアルゴリズム
E9	1E	カスタムアルゴリズム
F9	1F	このフェイスベクトルは2サイクルページバッファなし書込みコマンド、すなわち「バイト/語をプログラム」（コード10Hないし40H）及び3サイクルページバッファなし書込みコマンドすなわち「X8 8ビットで語をプログラム」（コードF0H）用に予約される

ページバッファ付き2サイクル書込み

【図21】



【図8】

コマンド コード	フェイス ベクトル	フェイス動作
9 8	0 9	このフェイスベクトルは2サイクルページバッファなし書き込みコマンド、すなわち「レディ/使用中フータチャクチャ・両端」(コード01H)用に予約されている
A 8	0 A	カスタムアルゴリズム
B 8	0 B	カスタムアルゴリズム
C 8	0 C	カスタムアルゴリズム
D 8	0 D	カスタムアルゴリズム
E 8	0 E	カスタムアルゴリズム
F 8	0 F	カスタムアルゴリズム
0 9	1 0	このフェイスベクトルは2サイクルページバッファなし書き込みコマンド、すなわち「NVロックビットセット」(コード11H)用に予約されている
1 9	1 1	カスタムアルゴリズム
2 9	1 2	カスタムアルゴリズム
3 9	1 3	カスタムアルゴリズム
4 9	1 4	カスタムアルゴリズム
5 9	1 5	カスタムアルゴリズム
6 9	1 6	カスタムアルゴリズム
0 8	0 0	このフェイスベクトルは3サイクルページバッファなし書き込みコマンド、すなわち「ページバッファからバイト/ワードをプログラム」コマンド(コード0C H)用に予約されている
8 9	1 8	カスタムアルゴリズム
9 9	1 9	このフェイスベクトルは2サイクルページバッファなし書き込みコマンド、すなわち「NVロックビットをロード」(コード01H)用に予約されている
A 9	1 A	このフェイスベクトルは2サイクルページバッファなし書き込みコマンド、すなわち「全チップ消去」(コードA7 H)用に予約されている

ページバッファ付き2サイクル書き込み

【図11】

コマンド コード	フェイス ベクトル	フェイス動作
8 A	0 2	「フェイスにより使用可能にしたテストモード」アルゴリズム。1番目のサイクルアドレスは不定値。データは2AH。2番目のサイクル:データはD0H(確認)。アドレスは不定値。3番目のサイクル:データはD0H(確認)だがアドレスは0P1E2H(「キー」として一部の注意の点)。3番目と3番目のサイクルのデータとアドレスはフェイスアルゴリズムによりコマンドを呼び出す「キー」として用いる
8 A	0 3	このフェイスベクトルは3サイクルページバッファなし書き込みコマンド、すなわち「ページバッファにアルゴリズムコード改訂IDをダンプ」用に予約されている
4 A	0 4	カスタムアルゴリズム
5 A	0 5	カスタムアルゴリズム
6 A	0 6	カスタムアルゴリズム
7 A	0 7	カスタムアルゴリズム
8 A	0 8	カスタムアルゴリズム
9 A	0 9	このフェイスベクトルは2サイクルページバッファなし書き込みコマンド、すなわち「レディ/使用中フータチャクチャの再構成」(コード01H)用に予約されている
A A	0 A	カスタムアルゴリズム
B A	0 B	カスタムアルゴリズム
C A	0 C	カスタムアルゴリズム
D A	0 D	カスタムアルゴリズム
E A	0 E	カスタムアルゴリズム
F A	0 F	カスタムアルゴリズム
0 B	1 0	このフェイスベクトルは2サイクルページバッファなし書き込みコマンド、すなわち「NVロックビット設定」(コード11H)用に予約されている
1 B	1 1	カスタムアルゴリズム

ページバッファなし3サイクル書き込み

【図15】

RDY/ BSY	SUSP	BPAIL	PFALL	LVPP	予約済	予約済	予約済	OSR6
OSRA7	OSRA6	OSRA5	OSRA4	OSRA3	OSRA2	OSRA1	OSRA0	OSR16A
UIで更新されるビット								
RDY/ BSY	SUSP	OPFALL	SLEEP	QFULL	PCRRODY	予約済	予約済	OSR10
UIでこのビットを更新								
RDY/ BSY	BLKLOCK	OPFALL	ABORT	QFULL	LVPP	予約済	予約済	OSR6

【図13】

コマンド コード	フェイズ ベクトル	フェイズ動作
0 C	0 0	ページバッファからバイト/語をプログラム。ページ バッファからプログラムするバイト数はコマンドの2 番目、8番目のサイクル中に与えられる
1 C	0 1	カスタムアルゴリズム
2 C	0 2	このフェイズベクトルは3サイクルページバッファな し書き込みコマンド、すなわち「フェイズアルゴリズム でテストモードを使用可能にする」コマンド（コード 2 A H）用に予約されている
3 C	0 3	このフェイズベクトルは3サイクルページバッファ付 き書き込みコマンド、「ページバッファ付アルゴリズム コード改訂1 Dをダンプ」用に予約されている
4 C	0 4	カスタムアルゴリズム
5 C	0 5	カスタムアルゴリズム
6 C	0 6	カスタムアルゴリズム
7 C	0 7	カスタムアルゴリズム
8 C	0 8	カスタムアルゴリズム
9 C	0 9	このフェイズベクトルは2サイクルページバッファな し書き込みコマンド、すなわち「フェイズ使用中アーキテ クチャの再構成」（コード9 6 H）用に予約されている
A C	0 A	カスタムアルゴリズム
B C	0 B	カスタムアルゴリズム
C C	0 C	カスタムアルゴリズム
D C	0 D	カスタムアルゴリズム
E C	0 E	カスタムアルゴリズム
F C	0 F	カスタムアルゴリズム
0 D	1 0	このフェイズベクトルは2サイクルページバッファな し書き込みコマンド、すなわち「NVロック設定」 （コード1 1 H）用に予約されている
F C	0 F	カスタムアルゴリズム

ページバッファ付3サイクル書き込み

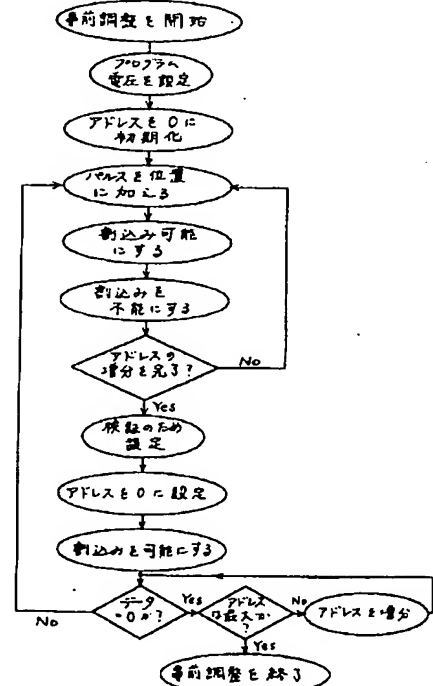
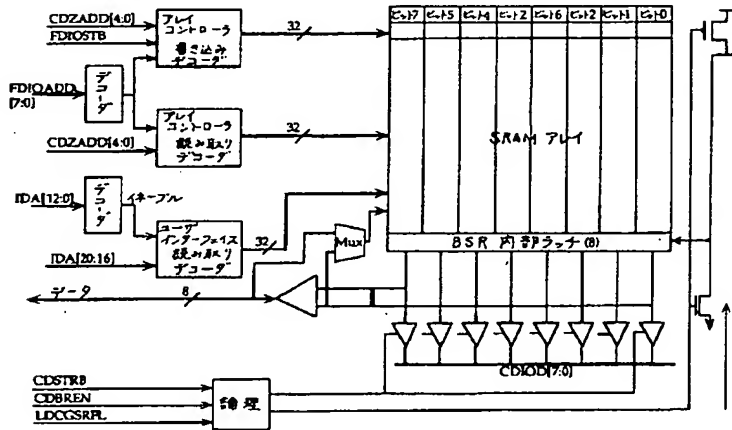
【図14】

コマンド コード	フェイズ ベクトル	フェイズ動作
2 D	1 2	カスタムアルゴリズム
3 D	1 3	カスタムアルゴリズム
4 D	1 4	カスタムアルゴリズム
5 D	1 5	カスタムアルゴリズム
6 D	1 6	カスタムアルゴリズム
P D	1 F	このフェイズベクトルは2サイクルページバッファな し書き込みコマンド「バイト/語をプログラム」コマン ド（コード10 H～10 H）及び3サイクルページバッ ファなし書き込みコマンド「18番目で語をプログラム」（ コードF B H）用に予約。両コマンドは同一フェイズ アルゴリズムをもちいる。」
8 D	1 8	カスタムアルゴリズム
9 D	1 9	このフェイズベクトルは2サイクルページバッファな し書き込みコマンド、すなわち「全チップ消去」（コード A 7 H）用に予約されている
A D	1 A	このフェイズベクトルは2サイクルページバッファな し書き込みコマンド、すなわち「NVロック解除」（コード A 7 H）用に予約されている
7 D	1 7	このフェイズベクトルは2サイクルページバッファな し書き込みコマンド、すなわち「単一ブロック消去」（コ ード10 H）用に予約されている
B D	1 B	カスタムアルゴリズム
C D	1 C	カスタムアルゴリズム
D D	1 D	カスタムアルゴリズム
E D	1 E	カスタムアルゴリズム

ページバッファ付3サイクル書き込み

【図22】

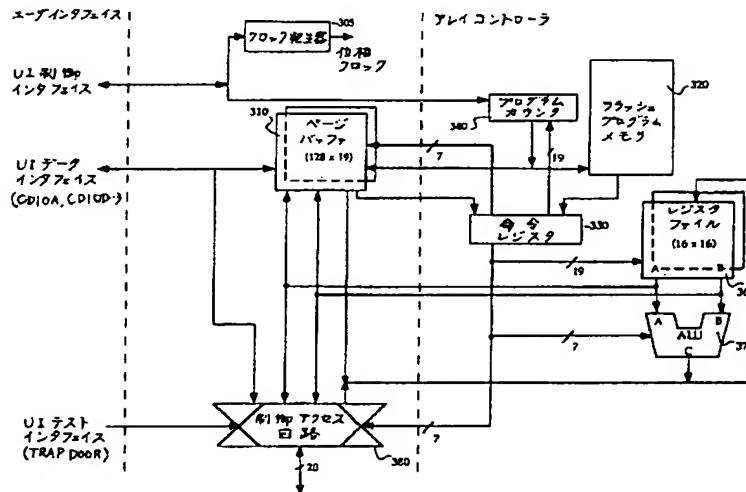
【図16】



【図17】

ビット	ニーモニック	定 義
7	DLI EOT/RESET	ブロックレディ/使用中。「0」はブロックが使用中であることを示す (BSR 特有)。パワーアップをリセット。(FUI は「1」でレディでフェイズは「0」を使い返る)
6	RV LOCK BIT	「0」パワーアップモーション。「0」はプログラム及び前走動作からブロックをロックする (BSR 特有)。パワーアップをリセット
5	OP FAIL	1 = このブロックの動作は失敗。(BSR 特有)。パワーアップをリセット
4	ABORT	1 = このブロックで動作は放棄。(BSR 特有) パワーアップをリセット
3	QFULL	1 = 動作待ち行列は一杯 (大域) (この設定が 0 にリセットされるまでユーザインターフェイスはこれ以上のコマンドを受けない)
2	LOW VPP	1 = ブロック上で作業しているときに低 VPP が検出される (BSR 特有)。パワーアップをリセット
1	RESERVED	今後の使用のために予約。値は保証されず、ユーザがマスクすべき。パワーアップをリセット
0	RESERVED	今後の使用のために予約。値は保証されず、ユーザがマスクすべき。パワーアップをリセット

【図18】



ドと実施されるアレイコントローラベクトルを例示した図表である。

【図15】 例示的な状態レジスタビット定義を示したブロック図である。

【図16】 本発明のブロック状態レジスタの構造を示したブロック図である。

【図17】 ブロック状態レジスタのビットの例示的な構成を示す。

【図18】 本発明の不揮発性メモリで利用するアレイコントローラのブロック図である。

【図19】 プログラムカウンタ回路を示すブロック図である。

【図20】 本発明の割込み過程の流れを示す図である。

【図21】 本発明の割込み過程の流れを示す図である。

【図22】 本発明の教示による消去過程中的のプログラムを示す流れ図である。

【図23】 本発明の教示による消去過程中的のプログラ

ムを示す流れ図である。

【符号の説明】

300 コンピュータシステム
302 CPU
304 主記憶装置
306 ユーザバス
310 フラッシュメモリシステム
20 フラッシュセルアレイ
30 読取り／書き込み経路回路
40 ユーザインターフェイス回路
50 アレイコントローラ
60 制御アクセス回路
70 ページバッファ回路
80-85 制御レジスタ回路
200 コマンド状態マシン
210 一時待ち行列
230 動作待ち行列
250 ページバッファカウンタ
260 状態レジスタ

フロントページの続き

(72)発明者 リチャード・ジョセフ・ドゥラント
アメリカ合衆国 95621 カリフォルニア
州・シトラス ハイツ・ローペ レイン・
ナンバー21・7733

(72)発明者 キース・フレドリック・アンダーウッド
アメリカ合衆国 95662 カリフォルニア
州・オレンジヴェイル・グリーンバック
レイン ナンバー165・9175

(72)発明者 ロドニイ・アール・ロズマン
アメリカ合衆国 95667 カリフォルニア
州・プレイサーヴィル・アロウビー ドラ
イブ・1212

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☒ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.